Proposal Number:

A Scientific Web-based Application for Global Tropical Cyclone Monitoring

FY 2002 Proposal to the NOAA HPCC Program

Principle Investigator: Dr. Mark D. Powell

Line Organization: OAR
Routing Code: R/E/AO/HRD

Address:
NOAA/AOML
Hurricane Research Division
4301 Rickenbacker Cswy
Miami, FL 33149-1097
Phone: (305) 361-4403
Fax: (305) 361-4402
E-mail Address: Mark.Powell@noaa.gov

Co-Investigator: Nirva Morisseau-Leroy
E-mail Address: Nirva.Morisseau@noaa.gov

Proposal Theme: Crisis / Disaster monitoring or response
Funding Summary: FY 2002 $110,000.

Authorizing Manager:

____ Date:____________

Kristina Katsaros, AOML Director

______________________________________ Date:___________

Cathy Steward, Financial Officer, AOML

______________________________________ Date:___________

Mark D. Powell, PI, Hurricane Research Division, AOML

A Scientific Web-based Application for Global Tropical Cyclone Monitoring

Proposal for FY 2002 HPCC Funding

Prepared by: Mark D. Powell and Nirva Morisseau-Leroy

Executive Summary:

The advent of the Internet has spawned a whole new generation of Web-based applications that allow dynamic interactions with databases. The benefits of a web-based approach include ease of maintenance and development, and centralized distribution of data services. Historically, technologies such as HTML and Common Gateway Interface (CGI) have provided a mechanism for distributing dynamic content over the Web. Web programming techniques have matured rapidly since then to provide enterprise-level functionality, reliability, and scalability in business and scientific applications. Promising and increasingly popular technologies for dynamic content generation include Java servlets and JavaServer Pages (JSPs). These new technologies are part of Java 2 Enterprise Edition (J2EE), which offers a powerful platform on which to build Web-based component applications. We will test the J2EE suite of products by transforming the award-winning Distributed Real-time Hurricane Wind Analysis System (H*WIND) system into a web-based application running over the AOML intranet. Depending on our results, subsequent year's investigation will evaluate the software over the Internet and test the application as a global tool for monitoring tropical cyclones with potential for use in the United Nations World Weather Research Program's International Tropical Cyclone Landfall Program as a Forecast Demonstration Project (FDP).

Problem Statement:

The HRD Real-time Hurricane Wind Analysis System (H*WIND) is a software component application consisting of a suite of scientific objects written in Java using Java Database Connectivity (JDBC) and SQLJ implementations. JDBC provides the capability of embedding dynamic SQL in Java programs whereas SQLJ allows the embodiment of static and dynamic SQL in Java programs. Today, H*WIND provides atmospheric scientists with the necessary tools to graphically interact with real-time atmospheric observations (from all over the world) and generate real-time analyses of tropical cyclone surface wind measurements. H*WIND was first demonstrated at the NOAA Tech 2000 Conference in October of 1999 where it won the "Best JAVA Implementation" award. The JAVA version of H*WIND is a research tool with operational capabilities. H*WIND is a distributed system that ingests realtime tropical cyclone observations measured by land-, sea-, space-, and air-borne platforms into an object relational database, adjusts them to a common framework, and then graphically displays the data relative to the storm with interactive tools so scientists can quality control, objectively analyze, and visualize the information. H*WIND's greatest strength is the ability to quickly evaluate and analyze observations from many diverse observation platforms. This is especially useful as new and unproven remote sensing platforms come online. This season H*Wind is undergoing evaluation this by HRD and NHC scientists conducting real-time hurricane wind analyses every 3-6 hours in concert with the National Hurricane Center's forecast and warning cycle. Wind analyses are made available to the public on HRD's website at http://www.aoml.noaa.gov/hrd/Storm_pages/wind.html. Analyses have proven useful to the insurance industry for evaluating proprietary loss models and are also used by the research community to provide forcing for storm surge and wave models.

In spring of 2001 H*WIND was rated by the National Hurricane Center as the highest priority research tool for transition to operations. As such, HRD was successful in obtaining support from the U. S. Weather Research Project's Joint Hurricane Testbed (JHT) for a two-year effort to transfer H*Wind to operations. In addition, FEMA and NESDIS are providing support to provide satellite sensed hurricane wind measurements to H*Wind and ultimately provide hurricane wind field information to FEMA's HAZUS model for use in estimating damage and losses from hurricane landfalls. In addition, we have been working informally to provide H*Wind to the Department of Defense Joint Typhoon Warning Center and to the WMO Tropical Cyclone Forecast Center for the Indian Ocean at the French colony of Reunion. We have also been working informally with GFDL to investigate using H*WIND analyses to initialize the GFDL hurricane prediction model and with the developers of the Aerosonde, an unmanned autonomous airborne vehicle being tested to fly into hurricanes and send wind measurements in real-time. The aerosonde is projected to be a possible future global tropical cyclone reconnaissance platform.

H*WIND is a multi-tier application consisting of a FAT front-client and two back-end engines distributed onto two Unix servers. In a fat client approach, all components live in the client with only the persistent domain data residing on the server. Whereas the distributed nature of H*WIND and use of JAVA have many advantages in terms of code reuse, hardware independence, and redundancy, the fat client approach has maintenance, distribution, and performance limitations. More importantly, in the fat client approach, the application components are integrated in a tightly manner. The tight coupling of the components may require recoding several client components when scientific logic and database processing change.

Information technology resources are extremely limited in government research laboratories, hence economical approaches to performance, data delivery, software maintenance, and distribution are valuable. In fat clients, most of the application or business logic is downloaded to the client. Thus, network traffic is a major burden. Even though the code is written in JAVA, distribution of the H*WIND application would require testing the software to ensure proper operation using various versions of the JAVA Virtual Machine on various operating systems and assisting remote users in getting the software to run properly. Since H*WIND accesses an Oracle8*i* database (soon to be upgraded to Oracle9*i* database), code distribution would require significant investments by recipients desiring their own database infrastructure. We have designed H*WIND as a global tropical cyclone wind field monitoring system. While the modern weather services of the U. S., Japan, Australia, and Europe have access to NOAA's environmental observations through sophisticated IT infrastructures, many developing nations in tropical cyclone areas have little access to these data and even if they did would require significant investment in hardware and software to make use of the data. H*WIND is designed to allow users to interact with and analyze observations without the throughput overhead of physically delivering the observations. Since the end result of the H*WIND system is a variety of graphical analysis products derived from observations (and not the observations themselves), FGDC metadata are not required.

The J2EE programming model (Perrone and Changanti 2000, Alur et al, 2001) provides the ability to develop and deploy applications that take advantage of a wide range of new and evolving technologies. Due to its support for distributed applications, J2EE is now being used by many private and government institutions. It solves many of the problems associated with fat clients by distributing data and software applications from a centralized data center over a wide-area network. The J2EE platform allows loose coupling of components where the fitting components can live separately in different tiers (multitier architecture), thus reducing the need of changing client’s code when scientific logic and database processing need to be modified. This model has the potential of making NOAA's data (and software to make use of the data) available to a wide variety of users including weather services of developing nations. The proposed HPCC research will test the feasibility of this model for scientific applications in a research and operational environment.

This research is consistent with the goal of NOAA's HPCC program to disseminate real-time and historical information to users more completely, in usable forms, and in a timely manner via the Internet. The analysis system will help to fulfill NOAA's strategic plan objective as research to advance short-term warning and forecast services. These uses of storm information are consistent with the findings of the World Disasters Report 1999 issued by the International Federation of Red Cross and Red Crescent Societies (EOS 1999), the National Science and Technology Council symposium "Real-time Monitoring and Warning for Natural Hazards" (EOS 1998), and recent comments by United Nations Secretary General Kofi Annan calling for a "culture of prevention" citing wars and natural disasters as "the major threats to security of individuals and human communities worldwide" EOS (1999). Our research goals are also consistent with the National Academy of Sciences (1996) report, "Computing and Communications in the Extreme" which identified challenges confronting crisis managers, including: 1) "need for cooperation among many different actors", 2) "need to rapidly identify, collect, and integrate crucial information about the developing situation", and 3) "capability to make projections and initiate actions in the face of an inevitable degree of uncertainty and incompleteness of information".

Proposed Solution:

To overcome fat client limitations and make H*WIND more widely available, we propose to follow the J2EE model to develop a Web-based version of H*WIND. A web application is generally composed of a variety of components as explained below. The Java component modeling used in H*WIND is described in Morisseau_Leroy et al, 1999, 2000, and 2001). Portions of the proposed H*WIND Web-based application (using the J2EE platform) are discussed in Morisseau_Leroy 2001.

The J2EE platform is designed to support multitier architecture. Based on the J2EE architecture, the Web-based version of H*WIND will consist of several components and containers. A container is a service that provides the necessary infrastructure and support for a component to exist and for the component to provide its own services to clients. The H*WIND's components and containers will be distributed in the following tiers:

Client tier The client tier will interact with the end users and display information from the system to the end users. In the J2EE platform, HTML and Java applets in a client container implement this tier.

Web tier The Web tier will generate the presentation logic and will accept user responses from the presentation clients (HTML or Java applet clients). Java servlets and JavaServer Pages (JSP) in a Web container will implement this tier.

Business or scientific tier This tier will handle the core business and scientific logic of the application. Enterprise JavaBeans (EJB) components in an EJB container, JavaBeans, and CORBA objects will implement this tier.

We propose also to investigate a Rapid Application Development (RAD) tool such as Web Objects from Apple Computer and/or Oracle9i JDeveloper for evaluation of various component strategies.

Analysis:

We want to produce a thin-client, multi-tiered application using the Java 2 Enterprise Edition (J2EE) product from Sun Microsystems. In a thin-client application, the logic to access remote objects and databases is removed from the client and moved to an application server or middle-tier server freeing the client from scientific logic and database processing that can be done more efficiently in the server. Application servers provide the basic structure required for developing and deploying multi-tiered enterprise applications. Another advantage of removing database processing from the client is the reduction of network traffic. With the release of J2EE, the combination of Enterprise JavaBeans (EJB), servlets, and JSP offers a powerful platform on which to build web-based component applications (Tait 2000). The EJB effectively encapsulates business and scientific logic while the JSP and servlet Web-tier provides the HTTP-based data entry and presentation to the ?front-end? for human interaction.

The advantages of an H*WIND Web-based application are manifold: The thin client architecture distributes much of the design to servers leaving as little processing code as possible on the client. The result is the reduction of network traffic, faster application downloads, and less client RAM. The thin client is also easier to maintain. With only one version distributed all over the world a wider pool of users can use H*WIND. This is especially important for potential use by developing nations and international regional tropical cyclone forecast centers. Thin Web-based client applications are more economical and efficient than fat, locally based clients. The Web-based version of the H*WIND system will rely on several cutting edge technologies including XML, JAVA servlets, JavaBeans, and JavaServer Pages. Any Web-based application requires the use of HTML, the only language that the Internet understands. HTML tells you how to format a document and does not give any information regarding the content of the document. HTML is used for designing static web pages. To overcome the limitation of HTML, we propose to use the Extensible Markup Language (XML). XML is a way of specifying the content elements of a page to a Web browser. Thus, XML tells you about the content of the document whereas HTML does not.

In multi-tier architecture, Java servlets provide web developers with a simple, consistent mechanism for extending the functionality of a web server and for accessing existing systems. Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs. We propose to remove from the H*WIND client all remote object invocation logic and move it to Java servlets. This should result in increased functionality, network traffic reduction, and better application and system performance.

JavaBeans component architecture extends "Write Once, Run Anywhere" capability to reusable component development. It is used for developing or assembling network-aware solutions for heterogeneous hardware and operating system environments -- within the enterprise or across the Internet. JavaBeans code runs on every OS and also within any application environment. In the Web-based version of H*WIND, we propose to use dynamic web pages versus static pages. JavaBeans allows us to do so. Java code to compute the dynamic content can be conveniently placed in one or more JavaBeans. This separation allows the Java programmer to focus on writing the computing logic in beans, and the HTML programmer to write the presentation format. JSP technology is an extension of the servlet technology created to support authoring of HTML and XML pages. It makes it easier to combine fixed or static template data with dynamic content. A JSP page may generate not just HTML but also XML output. JavaServer Pages (JSP) technology allows web developers and designers to rapidly develop and easily maintain information rich, dynamic Web pages that leverage existing systems. The JSP framework has been designed for use with modular and reusable software components such as JavaBeans. JavaBeans can easily be 'plugged into' a JSP for easy manipulation. The combination of JavaBeans and JSP provides clean separation of the Java logic that generates dynamic content from the presentation logic written in HTML.

Performance Measures:

Milestones:

Implementation of a Web-based Application requires two parts:

Part 1 (Months 0-6) Implementation of a back-end query engine. This engine will reside in an Application Server and will provide the interaction with the HURDAT historical storm track database.

A) Develop EJB components to encapsulate the query mechanisms to manipulate the HURDAT and H*WIND databases.

B) While developing the Java components, investigate the performance gain or loss in using JDBC and Oracle SQLJ implementations.

Part 2 (Months 7-12) Implement the Client and Web tier to interact with the back-end.

A) Develop Java servlets, JavaServer Pages (JSPs) and JavaBeans to communicate with the middle tier HURDAT and H*WIND EJBs and at the same time, to manipulate the data for the presentation layer.

B) Improve front-end client through use of Java applets and possibly XML.

Deliverables:

1. Conversion of the HURDAT file-based database to an Oracle object-relational database.

2. A Web application for querying HURDAT and H*WIND databases. This application will display Atlantic tropical cyclone tracks and H*WIND observations in response to user queries.

2001 HPCC Project Progress Report (Since receiving funds May 30 2001)

1. Developed a prototype back-end query engine to interact with the HURDAT storm track database.

2. Evaluated Oracle JDeveloper Rapid Application Development tool.

3. Developed five Java components to encapsulate the query mechanism for manipulating HURDAT storm tracks. This process investigated the distributed component technologies JavaBeans, EJB, and CORBA server objects.

Project related References

Alur, D., Crupi, J., Malks, D., 2001: Core J2EE Patterns: Best Practices and Designs. Sun Microsystems Press; ISBN 0130648841.

EOS, 1998: Real-time monitoring and warning for natural hazards can provide real-time benefits. AGU Transactions, p.329.

EOS, 1999: Report warns of more natural disasters. AGU Transactions, 317-319.

EOS, 1999: UN head calls for "culture of prevention" in addressing natural disasters. AGU Transactions, 438.

National Research Council, 1996: Computing and Communications in the Extreme.
Morisseau-Leroy, N., Solomon, M., and Momplaisir, G., 1999: Oracle 8I SQLJ Programming. Osborne McGraw-Hill; ISBN 0072121602.

Morisseau-Leroy, N., Basu, J., and Solomon, M., 2000: Oracle 8I Java Components, Osborne McGraw-Hill; ISBN 0072127376

Morisseau-Leroy, N., Solomon, M., and Momplaisir, G., June 2001: Oracle 9I SQLJ Programming, Osborne McGraw-Hill; ISBN: 0072190930

Perrone, P., Changanti, V., 2000: Building Java Enterprise Systems with J2EE. Sams Press; IBSN 0672317958

Tait, B., July 2000: Java Servlets and JSP: Separating Presentation from Business Logic. Java Developer Journal vol 5 issue 7 page 18

Budget Summary:

HPCC Contribution:

Category	Detailed Description	Amount
Personnel Compensation	Co. PI	$30K
Contract* Labor	Software Engineer	$30K
	Programmer	$25K
Capital Expenses	Computer Hardware/Software	$10K
Training/Travel	J2EE training	$15K
Total Requested		$110K

*Through contract to the NOAA-University of Miami Cooperative Institute for Marine and Atmospheric Science (CIMAS)

HRD Contribution:

PI Labor	(15%)	$25K

Co-PI Labor	(25%)	$30K
Software Engineer	(50%)	$30K
Programmer	(35 %)	$13.4K

Category	**Detailed Description**	**Possible**
2003 Request		**FY 2003**
Personnel Compensation	Co. PI	$35K
Contract* Labor	Software Engineer	$30K
	Programmer	$30K
Capital Expenses	Computer Hardware/Software	$7K
Training/Travel	J2EE training	$8K
Potential Funds To Be Requested For 2003		$110K

Budget Office Contact

- Administrative Officer: Cathy Steward
- Phone number: (305) 361-4303
- E-mail address: cathy.steward@noaa.gov
- FMC number: 940
